

09/730,190

MS160309.01/MSFTP170US

REMARKS

Claims 1-50 are currently pending in the subject application and are presently under consideration. Claim 43 has been amended herein to correct a minor informality. A version of all pending claims is found at pages 2-9. Favorable reconsideration of the subject patent application is respectfully requested in view of the comments herein.

I. Rejection of Claim 43 Under 35 U.S.C. §112

Claim 43 stands rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Withdrawal of this rejection is respectfully requested in view of the amendment made to claim 43 to rectify a minor informality.

II. Rejection of Claim 1 Under 35 U.S.C. §102(e)

Claim 1 stands rejected under 35 U.S.C. §102(e) as being anticipated by Sievert *et al.* (US 6,687,729). It is respectfully requested that this rejection be withdrawn for at least the following reasons. Sievert *et al.* does not teach or suggest each and every limitation as set forth in the subject claim.

For a prior art reference to anticipate, 35 U.S.C. §102 requires that "*each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)) (emphasis added).

The claimed invention relates to computer systems, and, more particularly, to systems and methods for implementing a client-side HTTP stack. (*See* pg.1, ln. 4-6). In a client/server system architecture, there exists a requesting machine (*e.g.*, the client) and a supplying machine (*e.g.*, the server) such that the client requests and receives data from the server in such a manner that provides for efficient handling of the received data. (*See* pg.1, ln. 15-20). As recited in independent claim 1, the invention provides *for a client side HTTP stack software component for processing requests, comprising: at least one*

09/730,190

MS160309.01/MSFTP170US

completion port object; a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request; and a client side state machine associated with the at least one request. It is apparent the claimed invention implements at least one completion port object along with the thread pool and state machine, to implement an HTTP stack that can efficiently handle HTTP requests on the client side of the client/server system architecture. The claimed invention is novel due to, but not limited to, the client-side nature of this stack implementation that can efficiently handle HTTP requests to a server by using multiple sockets and a thread pool (See pg.3, ln. 19-22). The claimed invention proposes the aforementioned implementation of this stack that allows the client to handle requests using a multi-threaded approach that was previously implemented exclusively on the server side of the processing schema. (See pg.2, ln. 14-25). Sievert *et al.* fails to teach or suggest such claimed aspects of the invention.

Sievert *et al.* relates to "threads of computer program execution" (See Sievert *et al.* col.1, ln.15-16) and a method for managing resources in order to facilitate the efficient handling of system calls using a system and method that implements a queue data structure in conjunction with a thread pool. Sievert *et al.* proposes an I/O completion port to which WIN32 API calls are provided (See Sievert *et al.* col. 3, ln.30-32) instead of a completion port *object* as recited in independent claim 1. The port mentioned in Sievert *et al.* utilizes a physical port through which data is processed, while the claimed invention proposes an object within a program to further functionality of the program.

Additionally, Sievert *et al.* fails to teach or suggest *a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request* as recited in the subject claim. Instead, Sievert *et al.* proposes a thread pool for "performing items of work" (See Sievert *et al.* col. 3, ln.21-22) that does not have any obvious applications associated with the processing of at least *one client side request*.

Furthermore, Sievert *et al.* fails to teach or suggest *a client side state machine associated with the at least one request*. Rather, Sievert *et al.* discloses a state machine that functions with respect to a work queue data structure implemented to facilitate the management of processing and threads. This state machine does not have any obvious

09/730,190

MS160309.01/MSFTP170US

applications to a state machine that is related to, or associated with, an HTTP request as recited in claim 1.

In view of at least the foregoing, it is apparent that Sievert *et al.* does not anticipate or suggest the subject invention as described in independent claim 1 (and claims that depend therefrom). Accordingly, this rejection should be withdrawn.

III. Rejection of Claims 8, 23, 35, and 46 Under 35 U.S.C. §102(b)

Claims 8, 23, 35, and 46 stand rejected under 35 U.S.C. §102(b) as being anticipated by IBM Technical Disclosure Bulletin ("Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls") (hereinafter IBM). It is respectfully requested that this rejection be withdrawn for at least the following reasons. The IBM Technical Disclosure Bulletin does not teach or suggest each and every limitation as set forth in the subject claims.

As discussed previously, the present invention relates to computer systems, and, more generally, systems and methods for implementing a client-side HTTP stack. (See pg.1, ln. 4-6). As recited in independent claim 8 (and similarly in independent claims 23, and 46), the invention relates to *a software component for implementing a client side HTTP stack comprising: a thread pool comprising N threads adapted to process M request from a client application component, wherein N and M are integers greater than 1 and wherein M is greater than N*. Further as recited in independent claim 35, the invention also pertains to *a computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N*. Specifically, the novelty of the claimed invention lies in the definition of this system as a client-side implementation for HTTP requests (See pg. 2, ln. 23-25). The references fails to teach or suggest such novel claimed aspects of the current invention.

Instead, the IBM Technical Disclosure Bulletin discloses an algorithm for determining the number of **executor** threads for a distributed computing environment (See IBM pg. 199, paragraph 1). Additionally, the method disclosed by this reference is applicable to an application server receiving Remote Procedure Calls from other entities

09/730,190

MS160309.01/MSFTP170US

rather than a client device making HTTP requests (*See IBM pg.199, paragraph 1*). In particular, the IBM Technical Disclosure Bulletin fails to teach or suggest a *client side HTTP stack* as recited in independent claim 8 (and similarly in independent claims 23, and 46) or a *computer-readable medium having computer-executable instructions for the processing of M requests* as recited in claim 35. Rather, the reference discloses an algorithm for “controlling the creation and destruction of executor threads of an application server.” (*See IBM pg.199, paragraph 1*). The algorithm defined in the reference functions on executor threads, which are utilized to facilitate the functionality of an application server; however, such threads are not of relevance in a *client-side* implementation designed to efficiently handle requests.

In view of at least the above, it is apparent that the IBM Technical Disclosure Bulletin does not anticipate or suggest the subject invention as recited in independent claim 8 (and similarly in claims 23, and 46) and in independent claim 35. This rejection should be withdrawn.

IV. Rejection of Claim 2 Under 35 U.S.C. §103(a)

Dependent claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Jones *et al.* (US 6,003,061). Jones *et al.* discloses a “thread scheduling facility” (*See Jones et al. col. 18*) to manage thread operations, however this object is not defined to function as a thread itself. The claimed invention implements a *scheduler thread* to activate objects that can effectively handle client-side requests. As discussed *supra*, Sievert *et al.* does not suggest the current invention with respect to the independent claim upon which this claim relies, and the differences between Sievert *et al.* and the claimed invention are not overcome by Jones *et al.* Therefore, it is respectfully requested that this rejection be withdrawn.

V. Rejection of Claim 3 Under 35 U.S.C. §103(a)

Claim 3 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Okano *et al.* (US 6,725,253). Okano *et al.* defines a DNS response thread, a DNS sorting table, and a path load measurement result thread that all function under the control of a controller (*See Okano et al. col. 12*). The

09/730,190

MS160309.01/MSFTP170US

controller directs DNS inquiries to distinct DNS sorting tables based on the current load on the system in order to augment the overall efficiency of the system. This system then generates a DNS response thread as a result of an inquiry to the DNS system. The claimed invention differs most notably from the prior art because it defines a DNS thread that is using information from a queue data structure to *perform the resolution and signal an event* upon completion (*See* p. 11, ln.24-25) instead of having some other object or process perform the name-to-IP address translation. As noted *supra*, Sievert *et al.* does not teach or suggest the current invention with respect to the independent claim upon which this claim relies, and the disparities between Sievert *et al.* and the invention as claimed are not overcome by Okano *et al.* Therefore, this rejection should be withdrawn.

VI. Rejection of Claim 4 Under 35 U.S.C. §103(a)

Claim 4 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Paxhia *et al.* (US 6,493,749). Paxhia *et al.* defines an “alarm thread” that “wakes up periodically and checks a list of timers” related to the processing being performed by active threads (*See* Paxhia *et al.* col.41). This thread looks for expired threads in the table that indicate that an operation has failed. The expired timer is subsequently removed from the list and the corresponding process is terminated. The timeout thread, as defined in dependent claim 4 of the claimed invention, selectively performs a timeout operation on a socket as a result of a timer in the list. It is not defined that the timeout thread look for expired timers in the list to indicate that a timer should be removed and a socket closed. As shown above, Sievert *et al.* does not suggest the current invention with respect to the independent claim upon which this claim depends, and the differences between the Sievert *et al.* and the claimed invention are not overcome by Paxhia *et al.* Accordingly, it is respectfully requested that this rejection be withdrawn.

VII. Rejection of Claim 5 Under 35 U.S.C. §103(a)

Claim 5 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Paxhia *et al.* (US 6,493,749) and further in view of Jones *et al.* (US 6,003,061). Jones *et al.* defines a “thread scheduling facility” (*See*

09/730,190

MS160309.01/MSFTP170US

Jones *et al.* col. 18) to manage thread operation, yet it is not disclosed that this “facility” is itself a thread object. The claimed invention implements a *scheduler thread* that activates objects in order to satisfy requests. The definition of the scheduler as a thread, distinguishes it from the scheduler as defined in the documents cited. As discussed previously, Sievert *et al.* does not teach or suggest the current invention with respect to the independent claim upon which this claim relies, and Paxhia *et al.* and Jones *et al.* fail to make up for the deficiencies of Sievert *et al.* Accordingly, withdrawal of this rejection is respectfully requested.

VIII. Rejection of Claim 6 Under 35 U.S.C. §103(a)

Claim 6 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Paxhia *et al.* (US 6,493,749) in view of Jones *et al.* (US 6,003,061) as applied to claim 5 above, and further in view of Okano *et al.* (US 6,725,253). Okano *et al.* defines a DNS response thread, a DNS sorting table, and a path load measurement result thread that all function under the control of a controller (See Okano *et al.* col. 12). The controller directs DNS inquiries to several DNS sorting tables based on the current load on the system then generates a DNS response thread as a result of an inquiry to the DNS system. The claimed invention, in contrast, defines a DNS thread that is using information from a queue data structure to *perform the [name-to-IP address] resolution and signal[s] an event* upon completion (See p. 11, ln.24-25). As discussed *supra*, Sievert *et al.* does not suggest the current invention with respect to the independent claim upon which this claim relies, and Paxhia *et al.* Jones *et al.* and Okano *et al.* fail to make up for the aforementioned deficiencies with respect to independent claim 1. Accordingly, reversal of this rejection is respectfully requested.

IX. Rejection of Claim 7 Under 35 U.S.C. §103(a)

Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Sievert *et al.* (US 6,687,729) in view of Paxhia *et al.* (US 6,493,749) as applied to claim 4, and further in view of Okano *et al.* (US 6,725,253). As noted *supra*, Sievert *et al.* does not teach or suggest the current invention with respect to the independent claim upon which this dependent claim relies, and Paxhia *et al.* and Okano *et al.* fail to make up for

09/730,190

MS160309.01/MSFTP170US

the deficiencies presented by Sievert *et al.* with respect to independent claim 1 from which claim 7 depends. Accordingly, withdrawal of this rejection is respectfully requested.

X. Rejection of Claims 9-13, 17-19, 24-28, 32-34, 36-39, and 47 Under 35 U.S.C. §103(a)

Claims 9-13, 17-19, 24-28, 32-34, 36-39, and 47 stand rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (US 6,687,729). Sievert *et al.* defines a work queue that manages pending work. One specific aspect of the work queue as disclosed in Sievert *et al.* is a "completion port" that refers to a physical port. The claimed invention however, implements a completion port *object* that functions as part of a program. Additionally, the thread pool manager recited in Sievert *et al.* fails to mention any sort of event or *completion packet*, as recited in the claims. Furthermore, the "state machine" in the claimed invention differs from that of the cited document, in that the claimed invention utilizes a state machine that functions with respect to individual threads, while the document cited requires the state machine to function on a work queue data structure.

The IBM Technical Disclosure Bulletin fails to mention the use of any key that uniquely identifies a particular request in a context related to the activation or processing of requests. The reference also fails to address the manner in which the claimed invention deactivates particular threads, since the deactivation operation is defined to drain the work queue of pending operations instead of *selectively deactivating* threads. As explained above, the IBM Technical Disclosure Bulletin does not suggest the current invention with respect to the independent claims from which these dependent claims rely, and the differences between this reference and the invention as claimed are not overcome by Sievert *et al.* Therefore, this rejection should be withdrawn.

XI. Rejection of Claims 14, 29, 40, and 48 Under 35 U.S.C. §103(a)

Claims 14, 29, 40, and 48 stand rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (6,687,729) as applied to claims 13, 28, 39, and 47 respectively, and further in view of

09/730,190

MS160309.01/MSFTP170US

Jones *et al.* (6,003,061). Jones *et al.* defines a “thread scheduling facility” (See Jones *et al.* Col. 18) to manage thread operations, however this object is not defined to be a thread itself. The claimed invention on the other hand implements a *scheduler thread* to activate objects that can effectively handle client-side requests. Furthermore, the scheduler implemented in Jones *et al.* creates “thread data structure[s]” (See Jones *et al.* Col. 19) instead of *activating an object* as recited in claim 48. As discussed *supra*, the IBM Technical Disclosure Bulletin does not teach or suggest the current invention with respect to the independent claims from which these claims depend, and the differences between IBM and the claimed invention are not overcome by Sievert *et al.* and Jones *et al.* Withdrawal of this rejection is therefore respectfully requested.

XII. Rejection of Claims 15, 30, and 41 Under 35 U.S.C. §103(a)

Claims 15, 30, and 41 stand rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (US 6,687,729) in view of Jones *et al.* (US 6,003,061) as applied to claims 14, 29, and 40 respectively, and further in view of Okano *et al.* (US 6,725,253). As discussed *supra*, the IBM Technical Disclosure Bulletin does not teach or suggest the current invention with respect to the independent claims upon which these claims rely, and the inconsistencies between the IBM Technical Disclosure Bulletin and the invention as claimed are not overcome by the combination of Sievert *et al.*, Jones *et al.* and Okano *et al.* Therefore, this rejection should be withdrawn.

XIII. Rejection of Claims 16, 31, and 42-45 Under 35 U.S.C. §103(a)

Claims 16, 31, and 42-45 stand rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (US 6,687,729) in view of Jones *et al.* (US 6,003,061) in view of Okano *et al.* (US 6,725,253) as applied to claims 15, 30, 41 respectively, and further in view of Paxhia *et al.* (US 6,493,749). Paxhia *et al.* defines an “alarm thread” that “wakes up periodically and checks a list of timers” to determine the status of threads that are currently being processed (See Paxhia *et al.* col.41). This “alarm thread” does not selectively time out threads as defined in claim 4, but rather looks for expired threads in the table that indicate

09/730,190

MS160309.01/MSFTP170US

that an operation has failed. The timer associated with a failed operation is removed from the list and the process terminated. The timeout thread, as defined in the claimed invention however, selectively performs a timeout operation on a socket as a result of a timer in the list, and it is not specified that the timeout thread look for expired timers within the list. As discussed *supra*, the IBM Technical Disclosure Bulletin does not teach or suggest the claimed invention with respect to the independent claims upon which these dependent claims rely, and the differences between this reference and the claimed invention are not overcome by Sievert *et al.*, Jones *et al.*, Okano *et al.* and Paxhia *et al.* Accordingly, reversal of this rejection is respectfully requested.

XIV. Rejection of Claim 20 Under 35 U.S.C. §103(a)

Dependent claim 20 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Jones *et al.* (US 6,003,061). Jones *et al.* defines a "thread scheduling facility" (See Jones *et al.* Col. 18) that manages thread actions, however the object itself is not defined to be a thread. The claimed invention on the other hand implements a *scheduler thread* to activate objects in response to client-side requests. As stated previously, the IBM Technical Disclosure Bulletin does not teach or suggest the invention as claimed with respect to the independent claim upon which this claim relies, and Jones *et al.* fails to overcome the deficiencies between the reference and the claimed invention. Therefore, this rejection should be withdrawn.

XV. Rejection of Claim 21 Under 35 U.S.C. §103(a)

Dependent claim 21 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Okano *et al.* (US 6,725,523). As stated *supra*, the IBM Technical Disclosure Bulletin does not teach or suggest the current invention with respect to the independent claim upon which this claim depends, and Okano *et al.* fails to make up for the deficiencies presented by the Bulletin. Accordingly, withdrawal of this rejection is respectfully requested.

09/730,190

MS160309.01/MSFTP170US

XVI. Rejection of Claim 22 Under 35 U.S.C. §103(a)

Dependent claim 22 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Paxhia *et al.* (US 6,493,749). It is respectfully requested that this rejection be withdrawn for at least the following reasons. Claim 22 depends from independent claim 8, and for reasons stated above, the IBM Bulletin does not teach or suggest all the limitations set forth in independent claim 8, and Paxhia *et al.* fails to rectify those deficiencies. Accordingly, withdrawal of this rejection is respectfully requested.

XVII. Rejection of Claim 49 Under 35 U.S.C. §103(a)

Dependent claim 49 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (US 6,687,729) as applied to claim 47, and further in view of Okano *et al.* (US 6,725,523). This rejection should be withdrawn for at least the following reasons. Claim 49 depends from independent claim 46, and as stated *supra*, Sievert *et al.* and Okano *et al.* fail to make up for the deficiencies presented by the IBM Technical Disclosure Bulletin with respect to independent claim 46. Accordingly, withdrawal of this rejection is respectfully requested.

XVIII. Rejection of Claim 50 Under 35 U.S.C. §103(a)

Dependent claim 50 stands rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin in view of Sievert *et al.* (US 6,687,729) as applied to claim 47, and further in view of Paxhia *et al.* (US 6,493,749). This rejection should be withdrawn for at least the following reasons. Claim 50 depends from independent claim 46, and neither Sievert *et al.* nor Paxhia *et al.* make up for the aforementioned deficiencies presented by IBM. Withdrawal of this rejection and allowance of claim 50 is respectfully requested.

09/730,190

MS160309.01/MSFTP170US

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicant's undersigned representative at the telephone number below.

Respectfully submitted,

AMIN & TUROCY, LLP



Himanshu S. Amin

Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731